Comparing models for Video-Based Speech Recognition

|  |  |  |  |
| --- | --- | --- | --- |
| Hyuntae Roh  *Viterbi School of Engineering* *University of Southern California* Los Angeles, the United States hroh@usc.edu | Natasha Denisyuk  *Viterbi School of Engineering* *University of Southern California* Los Angeles, the United States denisyuk@usc.edu | Yuko Ohomori  *Viterbi School of Engineering* *University of Southern California* Los Angeles, the United States yohmori@usc.edu | Isha Nitin Jain  *Viterbi School of Engineering* *University of Southern California* Los Angeles, the United States ishaniti@usc.edu |

This study evaluates the effectiveness of different Recurrent Neural Network architectures for video-based speech recognition. In particular, we compare LSTM, GRU, Bi-GRU, and Bi-LSTM models and their accuracy on the subsection of the GRID dataset. Our results demonstrate that Bi-LSTM achieves the best performance with a validation loss 2.927. Our findings show that using Bi-LSTM with speech recognition was the most effective across our experiments.

Keywords—Lipread, LSTM, GRU, Bi-LISTM, Bi-GRU

# Introduction

Lip reading is the process of interpreting spoken words by observing the movements of a speaker's lips, face, and tongue without relying on audio cues. Traditionally, this skill has been performed by professional lipreaders; however, advancements in artificial intelligence have enabled significant progress in automated lip reading, making it a valuable tool in various domains. One key application is in forensics, where audio may be unavailable or unclear in video evidence, such as security footage or dashcam recordings. Understanding what individuals are saying in such scenarios can uncover critical details for crime investigations or accident analyses. Another key application is for a communication aid, particularly when integrated into wearable technologies like the Apple Vision Pros or Meta glasses. These devices can leverage lip reading to display real-time captions on the user interface, enhancing communication for individuals with hearing impairments or those in noisy environments. Lastly, lip reading contributes to advancements in speech recognition research. In challenging auditory environments, such as concerts or sports events, visual input from lip movements can significantly improve voice recognition accuracy.

This paper builds on the findings of two significant studies in the field: "Lipreading with Long and Short Term Memory" by Wand et al. and "A Lip Reading Method Based on 3D Convolutional Vision Transformer" by Wang et al. Wand et al. demonstrated the feasibility of lip reading using temporal-based neural network architectures, LSTM, achieving better performance compared to earlier support vector machine (SVM)-based approaches. Second, Wang et al. introduced a novel method using a 3D convolutional vision transformer to capture both spatial and temporal information, further advancing the accuracy and robustness of automated lip reading systems. In this paper, we explore and compare multiple model architectures for enhancing automated lip reading.

# Related work

The paper *"Lipreading with Long and Short Term Memory"* (Wand et al., 2016) represents a significant milestone in the development of automated lip reading systems. The authors explore the feasibility of using temporal neural network architectures, specifically long short-term memory (LSTM) networks, to enhance the accuracy of lip reading models. Wand et al. argue that traditional methods, such as support vector machines (SVMs), rely heavily on handcrafted features, which limit their ability to effectively capture the temporal dynamics of lip movements. By contrast, LSTMs are well-suited for this task as they can model sequential dependencies in data, allowing the system to learn the temporal patterns associated with speech.

Wand et al. conducted their experiments using the GRID audiovisual corpus, a widely used benchmark dataset for speech and lip reading tasks. The dataset comprises video recordings from 19 speakers, each reciting 1,000 sentences, amounting to a total of 28 hours of video. The structure of each sentence is predefined, consisting of six components: a command, color, preposition, letter, digit, and adverb. This fixed structure results in a vocabulary of 51 distinct words. Each sentence in the GRID dataset has a fixed duration of three seconds, recorded at a frame rate of 25 frames per second, yielding a total of 75 frames per video. The videos are grayscale with a resolution of 360 × 288 pixels. To focus on the relevant region, a local 40 × 40 pixel window around the speaker’s lips was extracted.

Each frame is converted into a feature vector, which is then passed through the LSTM layers. These layers learn to recognize and predict the temporal relationships in the sequence, ultimately decoding the spoken words. The experimental results show that this approach outperforms traditional methods’ accuracy by at least 8%.

An important contribution of the study is the emphasis on scalability and generalizability. Wand et al. demonstrate that LSTM-based models can adapt to different speakers and conditions, a challenge that often hinders the effectiveness of lip reading systems. This adaptability makes their method a strong foundation for further research in automated lip reading.

Another study, “*A Lip Reading Method Based on 3D Convolutional Vision Transformer”* by Wang et al., introduced a novel architecture of 3D convolutional layers and vision transformers. Lip-reading tasks require effective extraction of subtle spatiotemporal features from video frames and modeling sequential dependencies movements of the lips and teeth to produce sound.

In their method, Wang et al. utilized 3D convolutions to extract local spatiotemporal features and transformers to capture global spatial correlations from continuous video frames. The features extracted by these layers were passed to a Bidirectional Gated Recurrent Unit (BiGRU) for sequence modeling, enabling the extraction of temporal dependencies. They also incorporated a Squeeze-and-Excitation (SE) structure in the convolutional embedding layer to enhance the representation of critical features while reducing irrelevant information.

This study achieved state-of-the-art performance on large-scale datasets, like the LRW dataset, with an accuracy of 88.5%.

# Key accomplishment

In this study, we explored multiple model architectures to evaluate their effectiveness in lip-reading tasks. In addition to the LSTM model used in the original work by Wand, we implemented and analyzed the performance of LSTM with convolutional layers, GRU with convolutional layers, and Bi-GRU with convolutional layers. Inspired by the work of Wang et al., we incorporated convolutional layers to capture spatiotemporal dependencies, further enhancing the models' ability to process video data.

1. Performance table

|  | Model Architecture | | | | |
| --- | --- | --- | --- | --- | --- |
| LSTMa | LSTMb | GRU | Bi-GRU | Bi-LSTM |
| Validation Loss | 49.191 | 15.774 | 4.785 | 4.098 | 2.927 |
| Epoch | 53 | 66 | 100 | 44 | 46 |

a. LSTM as in Wand et.al.

b. LSTM with convolutional layers

We planned to train each model over 100 epochs for comparison and record their respective validation losses and corresponding epochs. The results, as summarized table I, highlight that the Bi-LSTM model achieved the lowest validation loss of 2.927, outperforming the other architectures. This superior performance can be attributed to the bidirectional nature of Bi-LSTM, which processes sequences both forward and backward, thereby capturing a more comprehensive view of temporal dependencies. This capability is particularly advantageous for lip-reading, where understanding the context of both past and future frames is crucial for accurate speech prediction.

Additionally, our experiments showed that incorporating Conv3D layers in the LSTM architecture significantly enhanced its performance compared to Conv2D. This improvement can be attributed to Conv3D's ability to capture spatiotemporal features directly, which is vital for video-based tasks like lip-reading. This observation aligns with insights shared during the presentation, emphasizing the importance of modelling temporal dynamics alongside spatial features for improved performance.

Table II summarizes the performance of different models on a test input sequence: "bin red at S nine again" (derived from video-only input without audio). It demonstrates the progression of model accuracy as the architectures grow in complexity.

1. prediction results

| Real text: ‘*bin red at S nine again*’ | | |
| --- | --- | --- |
| **Models** | Loss | Prediction |
| LSTMa | 49.191 | ‘la re t ne an’ |
| LSTMb | 15.774 | ‘bin wre at nine again’ |
| GRU | 4.785 | ‘bin red at nine again’ |
| Bi-GRU | 4.908 | ‘bin red at nine again’ |
| Bi-LSTM | 2.927 | ‘bin red at nine again’ |

a. LSTM as in Wand et.al.

b. LSTM with convolutional layers

# Experimental setup

## Training dataset

The dataset used in this experiment consists of data from a single speaker, with 1000 video clips and their corresponding annotations from the GRIDS audiovisual corpus used by Wand. The videos are pre-processed using the same approach as the first paper by Wand. The load\_video() function processes video frames using OpenCV, converting them to grayscale and extracting the mouth region (frame[190:236, 80:220, :]) based on hardcoded coordinates, as all the videos follow the same layout. These frames are then normalized to ensure consistent input across the dataset. The load\_alignments() function extracts tokens from the alignment files, excluding silent frames, and converts them to numerical representations for training.

## Training configuration

The training configuration for the models is summarized in Table III. Additionally, the models use a dense output layer consisting of a fully connected layer that outputs class probabilities through a SoftMax activation function.

1. Training Configuration

| Parameter | Details |
| --- | --- |
| Optimizer | Adam optimizer |
| Learning Rate | 0.0001 |
| Loss Function | Custom CTCLoss (for sequence modeling) |
| Epochs | 100 |

CTC (Connectionist Temporal Classification) loss is a powerful loss function used in sequence-to-sequence tasks where the alignment between input and output sequences is not explicitly known, such as in lipreading or speech recognition. Unlike traditional loss functions, CTC can handle variable-length sequences and does not require pre-aligned data. It works by producing a probability distribution at each time step for all possible output tokens, including a special "blank" token that allows the model to skip certain input frames. CTC then calculates the total probability of all possible alignments that could yield the correct output sequence and minimizes the negative log probability of the true output, enabling the model to learn to map input sequences to output sequences effectively.

# Approach and Experiments

Our study focuses on evaluating different neural network architectures for video-speech recognition using the GRID corpus dataset. This dataset was curated explicitly for studies in lipreading. The dataset consists of recordings from 19 speakers, delivering 1,000 sentences each. The total size of vocabulary used by the speakers is 51 unique words. The dataset encompasses 28 hours of video footage with a resolution of 360x288 at 25fps.

As a starting point, we followed the methodology in *Lipreading with LSTM* (Wand et al.) as the foundation for our Long Short-Term Memory (LSTM) model design and implementation.  Like the paper, we use the GRID dataset with input consisting of 40x40 pixel grayscale frames over 75 timesteps. Due to the computational limitations, we focused on a smaller dataset section. We used recordings from a single speaker with 1,000 videos. We used the LSTM model as our baseline and experimented with adding 3D convolutional layers, and other models, like GRU, Bi-GRU, and Bi-LSTM models. Below are the detailed descriptions of the implementation and performance of each model.

## LSTM

A diagram of a software algorithm

Description automatically generated

1. LSTM Model Architecture

As in Wand's paper, the architecture consisted of two stacked LSTM layers, each with 128 units and orthogonal kernel initialization. Each LSTM layer was followed by a dropout layer to mitigate overfitting, a dense layer, and a SoftMax activation function. The dense layer weights had a uniform distribution with bounds ±0.05. This architecture diagram is presented in figure 1.

We used an Adam optimizer function with a learning rate of 0.001. Our function was CTC loss, implemented with tf.keras.backend.ctc\_batch\_cost function. The CTC loss calculates the number of correctly mapping a sequence of predictions to the target sequence.

The model was trained for 100 epochs and achieved a validation loss 49.191 during training. The model's poor performance can be attributed to several factors. First, we flatten the input frames into 1600-dimensional vectors, which discard spatial relationships, such as the lips, teeth, and tongue position. This information is crucial for the lipreading task. Without convolutional layers to extract spatial features, the LSTM could not learn spatial and temporal dependencies from raw pixel data. Furthermore, training on only 1,000 clips from one speaker, as opposed to the full GRID dataset as done by Wand et. Al constrained model's capacity to learn effectively.

## LSTM with 3D Convolutional Layers

A diagram of a flowchart

Description automatically generated

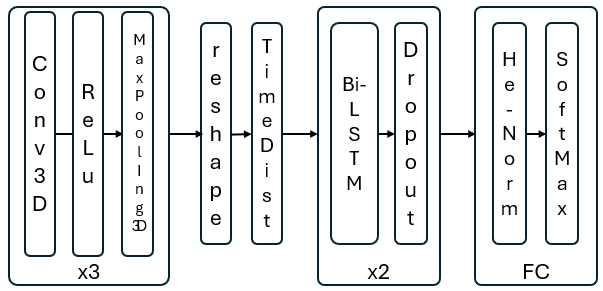
1. LSTM with 3D Convolutional Layers Architecture

As stated, the LSTM model performance may have been hindered by the lack of spatial relationships. To address this, we experimented with adding 3D convolutional layers. Inspired by the paper *A Lip Reading Method Based on 3D Convolutional Vision Transformer* (Wand et al.), we incorporated 3D convolutional layers. Adding convolutional layers allowed the model to capture local spatial patterns, such as the shapes associated with lip movements, before passing this information with these dependencies onto the LSTM layers. This approach aimed to retain crucial spatial relationships lost in the initial base model.

 The architecture diagram is presented in figure 2. The first layers in the architecture are the two 3D convolutional layers that capture patterns essential for lip reading movements. Each of the 3D convolutional layers was followed by a 3D max-pooling layer to extract spatiotemporal features. After the convolutional layers, the output was reshaped into a sequence of feature vectors to serve as input to the two stacked LSTM layers with orthogonal kernel initialization. After each LSTM layer, we had dropout layers to prevent overfitting. Finally, a TimeDistributed dense layer with softmax activation function was added last to provide predictions for each frame, aligning them with the target sequences.

The model was trained using the Adam optimizer with a learning rate of 0.001 and the CTC loss function. By adding these convolutional layers, the model could efficiently process spatial patterns before the temporal model. This improved the performance significantly compared to the base model, archiving performance of 15.774 at epoch 66

## Bi-directional LSTM



1. (Bi-LSTM) Model Architecture

Followed by Long Short-Term Model(LSTM) with convolutional layer, we tried Bi-directional Long Short-Term Model. The model design, in the Figure 3, has three sets of convolutional 3D layer, followed by ReLu Activation function and MaxPooling3D. Remind that the shape of input image data was (75, 46, 140, 1). The convolutional layer outputs 128 filters that has identical spatial dimensions as input. Maxpooling3D reduces spatial resolution from (46x140) to (23x70) while maintaining the temporal dimension unchanged.

The next step is reshaping and time-distributed flattening. For reshaping, we used keras embedded in tensorflow to flatten spatial dimension into a single vector for each frame. TimeDistributed module ensures that each time step is flattened independently.

After flattening, two sets of Bi-directional LSTM model was applied. Bi-LSTM model processes temporal dependencies,i.e., pattern across frames, in both forward and backward directions. Dropout layer with 50% rate was applied as a preventive measure of overfitting.

A graph of a training and validation loss

Description automatically generated

1. (Bi-LSTM) Training and Validation Loss over Epoch

The training was done 100 epoch, and the declining training and validation loss were observed as in the Figure 4. The lowest validation was 2.927 at 46th epoch.

## GRU

A diagram of a process

Description automatically generated

1. (GRU) Model Architecture

Followed by Bi-Directional LSTM (Bi-LSTM), we tried Gated Recurrent Unit(Bi-GRU) Model. The model design, in Figure 5, is similar to the previous designs. It has three sets of convolutional 3D layer, followed by ReLu Activation function and MaxPooling3D. Input image data was (75, 46, 140, 1). The convolutional layer outputs 128 filters that has identical spatial dimensions as input. Maxpooling3D reduces spatial resolution from (46x140) to (23x70) while maintaining the temporal dimension unchanged.

The next step is reshaping and time-distributed flattening. For reshaping, we used keras embedded in tensorflow to flatten spatial dimension into a single vector for each frame. TimeDistributed module ensures that each time step is flattened independently.

After flattening, two sets of Bi-directional GRU model was applied. Bi-GRU model captures temporal dependencies in both forward and backward directions. Since GRU is a lightweight recurrent unit compared to LSTMs load is relatively smaller. Dropout layer with 50% rate was applied.

A graph with blue and orange lines

Description automatically generated

1. (GRU) Training and Validation Loss over Epoch

The training was done over 100 epochs initially, and the declining training and validation loss were observed as in the Figure 6. The lowest validation was 4.785 at the 100th epoch.

A graph with blue and orange lines

Description automatically generated

1. (GRU) Training and Validation Loss over Epoch

Since the validation loss continued to reduce till the 100th epoch, the model was trained until an increase in validation loss was observed. The graph for the same is shown in Figure 7. The minimum validation loss achieved was 3.42 at epoch 119.

## Bi-directional GRU

A diagram of a diagram

Description automatically generated

1. (Bi-GRU) Model Architecute

Followed by Gated Recurrent Unit(GRU), we tried Bi-directional Gated Recurrent Unit(Bi-GRU) Model. The model design, in Figure 8, is similar to the previous designs. It has three sets of convolutional 3D layer, followed by ReLu Activation function and MaxPooling3D. Input image data was (75, 46, 140, 1). The convolutional layer outputs 128 filters that has identical spatial dimensions as input. Maxpooling3D reduces spatial resolution from (46x140) to (23x70) while maintaining the temporal dimension unchanged.

The next step is reshaping and time-distributed flattening. For reshaping, we used keras embedded in tensorflow to flatten spatial dimension into a single vector for each frame. TimeDistributed module ensures that each time step is flattened independently.

After flattening, two sets of Bi-directional GRU model was applied. Bi-GRU model captures temporal dependencies in both forward and backward directions. Since GRU is a lightweight recurrent unit compared to LSTMs, the computational load is relatively smaller. Dropout layer with 50% rate was applied as a preventive measure of overfitting.

A graph of a training and validation loss

Description automatically generated

1. (Bi-GRU) Training and Validation Loss over Epoch.

The training was done 100 epoch, and the declining training and validation loss were observed as in the Figure 9. The lowest validation was 4.098 at the 44th epoch.

# CONCLUSION AND FUTURE PLANS

The proposed GRU-based model demonstrates effective performance on the current dataset, which consists of a 51-word vocabulary. Despite limitations in the dataset and the short sequence structures, the model shows promising results. The combination of 3D convolutions and GRUs for spatial and temporal feature extraction, respectively, proves to be an effective architecture for video-based sequence learning tasks.

Future work aims to address the current dataset and model limitations. Expanding the corpus to include longer sequences and a larger vocabulary will provide a more comprehensive evaluation of the model’s capabilities. Additionally, variations in weight initialization methods could be explored to improve convergence speed and accuracy during training.

In terms of advanced models, integrating transformers and implementing beam search decoding strategies could significantly enhance sequence prediction accuracy. Furthermore, incorporating computer vision techniques for dynamic mouth tracking could improve the model’s ability to generalize across diverse datasets and applications.

##### References

1. Wand, M., Koutník, J., & Schmidhuber, J. (2016). Lipreading with long short-term memory. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6115–6119. <https://doi.org/10.1109/ICASSP.2016.7472852>
2. Wang, H., Pu, G., & Chen, T. (2022). A lip reading method based on 3D convolutional vision transformer. *IEEE Access*, 10, 77205–77212. <https://doi.org/10.1109/ACCESS.2022.3193231>